Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 35 (2014) 464 – 473
Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Extraction Japanese slang from weblog data based on script type and stroke count

Kazuyuki Matsumoto*, Kyosuke Akita, Xielifuguli Keranmu, Minoru Yoshida, Kenji Kita

The University of Tokushima, Minami Josanjima-cho 2-1, Tokushima City, 770-8506, Japan

Abstract

Young people commonly use slang in the texts for weblogs or Social Networking Sites. How to treat such slang words properly is one of the problems in the field of text mining. In this paper, we examined several methods to extract Japanese slang called "*Wakamono Kotoba*," which is particularly used by young people, by focusing on its script type and stroke count. In the evaluation experiment, a high precision was obtained when we adopted script type for extraction.

Peer-review under responsibility of KES International.
Keywords: Japanese slang; *Wakamono Kotoba*; Conditional Random Fields(CRF); script type; stroke count.

1. Introduction

Recently, availability of smart phones and tablet type PCs has expanded with an explosive pace, increasing the number of users of Social Networking Site (SNS), bulletin board and weblog sites. Smart phone is being used particularly by young people from the teenagers to those in their thirties, making weblog or SNS more popular among those young people.

Young people tend to use unique words distinctive to their generations (i.e. "*Riaju* (having fruitful life)", "*Komyusho* (having trouble in communication)," etc.), which are called *Wakamono Kotoba* in Japanese. They often use such *Wakamono Kotoba* also in the communication on weblog or SNS. Because *Wakamono Kotoba* is a word closely connected with daily life and is subject to change of trend, it is generated daily but falls into disuse in short period. Therefore, it is difficult to construct a dictionary that covers *Wakamono Kotoba*. Besides, some *Wakamono Kotoba* are used only among a limited communication group with a specific interest. And, people who do not belong to the communication group sometimes cannot understand the meaning of *Wakamono Kotoba* used in the group.

In the recent text mining technique that uses a collective wisdom, social data on Web are being used to identify human relationships or to conduct opinion mining or reputation analysis. However, in these data, slang is often included. It is considered that slang causes errors to the basic language analysis tools for morphological analysis,

* Corresponding author. Tel.: +0-88-656-7654.
E-mail address: matumoto@is.tokushima-u.ac.jp

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.127

syntactic parsing or semantic analysis, badly influencing the mining accuracy. In this paper, we focused on Japanese slang of *Wakamono Kotoba* on Web. We discussed on usage and application of *Wakamono Kotoba* in text mining, and proposed a method to extract *Wakamono Kotoba* automatically from a text data.

There were various approaches that have been made to extract automatically unknown words such as slang. Asahara et al.[4] proposed the character-based method to identify Japanese unknown word. Ling et al.[5] also proposed the character-based tagging and chunking method for Chinese unknown word. Ritter et al.[6] identified named entity by using LabeledLDA. Tsuchiya et al.[7] proposed the judgment method of alphabet abbreviation word that are not registered into a dictionary by using association mechanism.

Hadi et al.[8] constructed a slang dictionary for opinion words, and, Taysir et al.[9] also constructed an emotion polarity dictionary by automatically extracting slangs from arabic text using Support Vector Machine. Murawaki et al.[10] researched how to acquire unknown words automatically without manual supervision and register these words into a morphological analysis dictionary.

Many researchers studied on the problems of unknown word. However, there are a few research aiming to extract slangs using slang-specific features like the stroke numbers. In this paper, we report the effect of using such slang-specific features on the accuracy of slang extraction.

2. Japanese slang(*Wakamono Kotoba*)

Wakamono Kotoba(*WK*) is a Japanese language particularly used among the younger generation from teenagers to those in their late twenties[1]. They tend to use *WK* especially on Internet. Some examples of *WK*, their original words and meanings are listed below.

- *Kebai* . . . *Kebakeashii* (floozy)
- *Kuripa* . . . *Kurisumasu party* (Christmas party)
- *Dotakyan* . . . *Dotanba Cancel* (last-minute cancellation)

As described in the previous section, *WK* is sometimes limitedly used among a certain group of communication. There are varieties of such groups and varieties of WK used in the group, making it very difficult to collect all *WK*.

There are a lot of researches on how to process unknown words. However, there are a few that focused on *WK* because of the following reasons:

1. Varieties of notations
2. Short duration of use

It is hard to obtain its sense or usage because *WK* has many different notations and it is used relatively for short period. As the result, it is difficult to create a dictionary of *WK* by manually.

Matsuo et al.[2] proposed a method to extract *WK* written in *Katakana* by preparing a template: "word prior to *WK*" + "*WK*" + "word after *WK*," and by matching words with the template. As the result of the evaluation experiment, this method obtained 42.4% accuracy of extraction of *WK* written in *Katakana*. They analyzed the randomly chosen 217 extracted *WK* and found that 45 unknown *WK* had been successfully extracted. However, in the character strings that they analyzed, spelling mistakes and proper nouns such as personal name were included 57.6%. They concluded that they need to reduce the noise such as proper noun or spell-miss and to consider template using sentence structure.

Mori et al.[3] assumed that if the words have the same part of speech, the words have similar character strings before and after the words. Based on this assumption, they proposed a method to extract a word from a corpus and estimate its part of speech at the same time. This method defined the environment of part of speech and character string as conditional probability distribution of backward and forward character strings of the target character string in the corpus. In the experiment, by considering frequency of the character strings, the precision of 96.8% was obtained when threshold was set at 0.1. Without considering frequency of character string, the precision of 86.2% was obtained. The method successfully extracted 268 unknown words and proved its effectiveness.

Because the method by Matsuo et.al only focused on *WK* written in *Katakana* and used template matching for word extraction, its application is limited and cannot apply to new expressions. The problem of the method by Mori et al. is that it cannot deal with unknown words exclusively written in *Hiragana*.

3. Proposed method

When a sentence including *WK* is analyzed by the existing morphological analysis systems, analysis error will be caused and segmentation will not be appropriately made. This error happens because *WK* is not registered in the dictionaries for morphological analysis, and *WK* tends to be used in the sentences including casual expressions that do not follow grammatical rules.

This paper proposes a method to extract *WK* by using surface features of characters without using morphological analysis. We assumed that *WK* should be created by using daily words, and based on this assumption, we tried to extract *WK* focusing on its script type or stroke count by calculating Conditional Random Fields (CRF).

3.1. Surface feature of character string

Script type and stroke count are surface features that character string has. In this section, we explain about script type and stroke count, and consider the relation between these surface features and *WK*.

Firstly, we examined *WK* included in *WK* Emotion Corpus [11,12,13,14], defined several combination patterns of script types, then, for each combination pattern, we calculated its appearance frequency. Fig.1 shows the result. The horizontal axis shows combination patterns of script types that were used for *WK*, and the vertical axis shows the appearance frequency of them. In the figure, following abbreviations were used for each script type.

- Kata. : *Katakana* character
- Hira. : *Hiragana* character
- Chi. : Chinese character
- Eng. : Alphabet character
- Sym. : Symbol

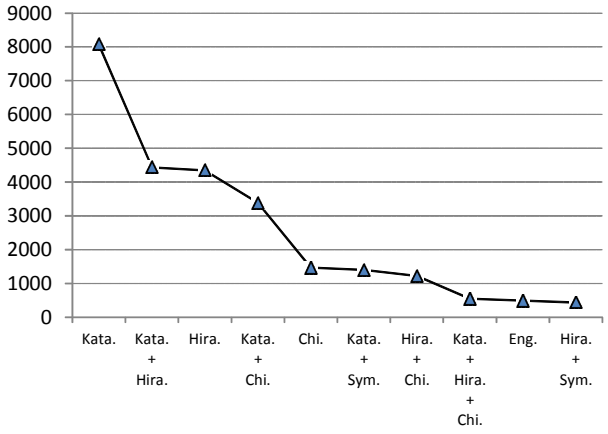

Fig. 1. The appearance frequency for each combination pattern of script types.

As in the figure, *WK* exclusively consisting of *Katakana* appeared most frequently. From these characteristics, we considered that script types should become a clue to extract *WK*. The stroke count is one of the information on a character, and the stroke count indicates the number of lines and dots consisting of a character. In Japanese language, the number of stroke counts sometimes affects the visual impression of a character such as being casual or formal. It is common for Japanese parents to name their children by considering stroke counts of Chinese character to use. This distinctively reflects that stroke counts can provide various impressions on characters. *WK* are rarely used in formal occasion but often used in relaxed and casual dialogue. Therefore, the characters that have easy and soft impression with less stroke counts seem to be used.

3.2. Conditional Random Fields(CRF)

Conditional Random Fields (CRF) is a method that applied a log-linear model to the sequential labeling problem. The sequential labeling problem is to output the appropriate label list Y when the label list X is given. CRF is one of the machine learning methods such as Hidden Markov Model or Support Vector Machine. The characteristics of the method is that it can obtain the output with probability value(probabilistic model) and that the output has a structure (structural learning) by learning the output list Y from the input list X. CRF is used to estimate part of speech or to extract unknown word in morphological analysis. CRF is also used for a morphological analysis tool called Mecab. The CRF analysis tools are as follows.

- Conrad[1]
- HCRF library(including CRF and LDCRF)[2]
- CRF++[3]
- CRFsuite[4]

CRF++ can use a training algorithm of MIRA(Margin Infused Relaxed Algorithm). And its training speed is higher than other tools. In this paper, we used CRF++ as a CRF analysis tool.

4. Experiment

The proposed method was evaluated by experiment. The training data was split into unit of character and each character was added following five types of labels.

B: Beginning of *WK*
I: Intermediate of *WK*
E: End of *WK*
S: *WK* as Single Word
O: Other than *WK*

Besides, information on script type and stroke count was annotated. Thus, each sentence d_i in the training and test data are expressed as a list of 4–tuples (c_i, k_i, s_i, l_i) where c_i is the i–th character, k_i is the script type, s_i is the stroke count, and l_i is the label. Fig. 2 shows the examples of the training data and the test data.

Recall, precision and F1-Score were calculated from both the correct labels of test data and the estimated labels by CRF, then the results were examined based on F1-Score. The equations to calculate Recall, Precision and F1-Score are shown in Equation 1, 2, 3. In these equations, R indicates the number of the character strings that were estimated correctly, C indicates the number of *WK* in the test data, and N indicates the total number of the estimated character strings.

[1] http://www.broadinstitute.org/annotation/conrad/
[2] http://sourceforge.net/projects/hcrf/
[3] http://crfpp.googlecode.com/svn/trunk/doc/index.html
[4] http://www.chokkan.org/software/crfsuite/

Char. No.	Character	Type	Stroke	Label
1	ヤ	Kata.	2	B
2	フ	Kata.	1	I
3	オ	Kata.	3	I
4	ク	Kata.	2	E
5	で	Hira.	4	O
6	も	Hira.	3	O
7	一	Chi.	1	O
8	切	Chi.	4	O
9	見	Chi.	7	O
10	掛	Chi.	11	O
11	け	Hira.	3	O
12	ま	Hira.	4	O
13	せ	Hira.	3	O
14	ん	Hira.	2	O

Fig. 2. The example of training data and test data.

$$\text{Recall} = \frac{R}{C} \tag{1}$$

$$\text{Precision} = \frac{R}{N} \tag{2}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

In this paper, the test data and the training data did not have a common *WK* because our aim was to extract new *WK* and estimate the position of *WK*. In the experiment, following three types of features were used for the training data and the test data.

- Use only character surface as feature (C)
- Use character surface and script type as feature (C+K)
- Use character surface, script type and stroke count as feature (C+K+S)

We prepared three kinds of training data consisting of 500 sentences, 1,000 sentences and 5,000 sentences. The results were calculated for each training data and how the results changed according to the number of the training data was examined. Three kinds of templates of feature were prepared for CRF++ and experiments were conducted with each template. An example of feature template is shown in Fig. 3.

In below, how to interpret the feature template is explained. For example, 'U00 %x[-2,0]' indicates the feature 'C' that locates two features prior to the *i*-th feature in the target sentence. As 'U07' in the figure, it is possible to use a combination of multiple features as a feature.

We tried the following three optional regularizations for CRF++.

- L1-regularization
- L2-regularization
- MIRA(Margin Infused Relaxed Algorithm)

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-2,0]/%x[-1,0]/%x[0,0]
U06:%x[-1,0]/%x[0,0]/%x[1,0]
U07:%x[0,0]/%x[1,0]/%x[2,0]

U08:%x[-2,1]/%x[-1,1]/%x[0,1]
U09:%x[-1,1]/%x[0,1]/%x[1,1]
U10:%x[0,1]/%x[1,1]/%x[2,1]

U11:%x[-2,0]/%x[-2,1]/%x[-2,2]
U12:%x[-1,0]/%x[-1,1]/%x[-1,2]
U13:%x[0,0]/%x[0,1]/%x[0,2]
U14:%x[1,0]/%x[1,1]/%x[1,2]
U15:%x[2,0]/%x[2,1]/%x[2,2]

# Bigram
B
```

Fig. 3. A part of feature template of "C+K+S."

In the experiment, we used MIRA that had obtained the highest accuracy. MIRA[15] is one of the online learning methods and has been used in the existing research such as parsing[15] and morphological analysis[16]. The default value was used in other optional parameters. In the experiment, we prepared three kinds of training data consisting of 500, 1,000 and 5,000 sentences that were extracted from *Wakamono Kotoba* Emotion Corpus(WKEC)[11,12,13,14]. The number of *WK* included in each training data are indicated in Table 1.

Table 1. The number of *WK* included in the training data.

Training Data Size	# of *WK*
500	596
1,000	1,200
5,000	6,037

The test data was also extracted from the WKEC. 200 conversational sentences and 278 *WK* were included in the data, and their script types, stroke counts and output labels were annotated on character basis as shown in Fig.1. Table 2 shows frequency of *WK* included in the test data according to the script type patterns.

Table 2. Frequency of *WK* according to the script type patterns in the test data.

Pattern	Freq.(%)
Kata.	99 (35.612)
Hira.	48 (17.266)
Kata. + Hira.	39 (14.029)
Kata. + Chi.	33 (11.871)

5. Result and discussion

Fig. 4 shows the results of the experiments. The horizontal axis indicates the combination of features and the vertical axis indicates F1-Score. The improvement of the score was recognized by the proposed method in Fig.4.

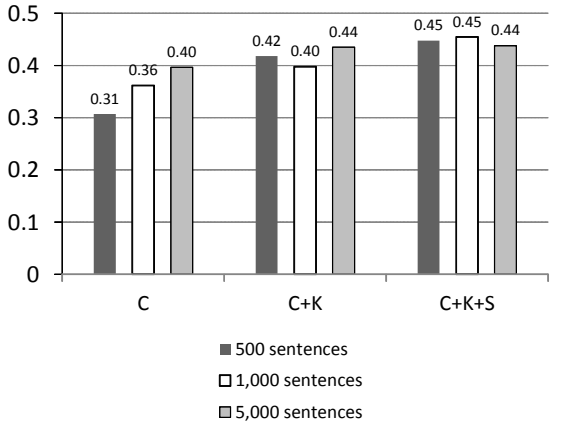

Fig. 4. The F1-Score according to each training data size.

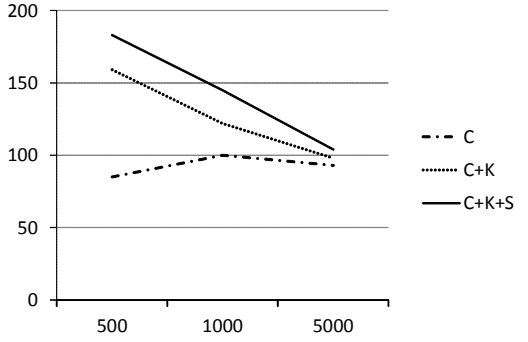

Fig. 5. The number of the extracted character strings in each training data size.

However, when the size of training data became larger, the difference between “C+K” and “C+K+S” increased, and when data size was 5,000 sentences, there were almost no differences in the scores. Fig.5 shows the number of the extracted character strings by the proposed method.

When the size of training data size was 500, “C+K+S” could extract twice as many as character strings than “C” could. Table 3 shows the recalls and the precisions of the experimental result.

Successful examples and failed examples are shown in Table 4,5. In Table 4, the successfully extracted *Wakamono Kotoba* are indicated with underline, and in Table 5, the estimated character strings are indicated with underline, and the correct character string is indicated in brackets .

Table 6 shows the results obtained by analyzing the character strings that were introduced as examples in Table 4 and Table 5, by Mecab with CRF. ‘T’ indicates the successful examples, and ‘F’ indicates the failed examples.

Table 3. Recall and Precision.

Size		C	C+K	C+K+S
500	Recall	0.20	0.34	0.37
	Precision	0.62	0.55	0.54
1,000	Recall	0.25	0.29	0.35
	Precision	0.66	0.63	0.65
5,000	Recall	0.27	0.30	0.31
	Precision	0.76	0.81	0.77

Table 4. Successful examples.

Abura mo , Yabai
Kishoi ! Kishosuguru
Jikochu de Jiishikikanjo de
Shinaitame nimo Hadekon wo sureba
Nikonama wa genjitsusekai de aite ni
Dotakyan sarete shimaimashita

Table 5. Failed examples.

Extraction Part	*Wakamono Kotoba*
Gejun ni inpark shiyouto	*inpark*
Nakanaka eguishi nikata wo	*egui*
Turuturu no abura mo	*Turuturu*
Soreni kabegami mo getto dekiruyou	*getto*
Choko wo pakegai saseru	*pakegai*
Hadekon banzai !!	*Hadekon*
Yuushuu puchipura cheak desu	*puchipura*

Table 6. Split example by Mecab.

T/F	Example
T	*Abura / mo / , / Yabai*
F	*Kishoi / ! / Kisho / suguru*
F	*Jiko / chu / de / jiishiki / kajo / de*
F	*Shi / nai / tame / ni / mo / hade / kon / wo / sure / ba*
F	*Niko / nama / wa / genjitsu / sekai / de / aite / ni*
T	*Dotakyan / sa / re / te / shimai / mashi / ta*
F	*Gejun / ni / in / park / shiyo / u / to*
F	*Nakanaka / egu / ishi / ni / kata / wo*
T	*Sore / ni / kabegami / mo / getto / dekiru / you*
F	*Moripapa / ! / hade / kon / banzai / !!*
F	*Yuushu / puchipura cheak / desu*

With the proposed method, even though some *WK* (i.e. "*Hadekon*") were successfully extracted from some sentences, they failed to be extracted from other sentences. This failure was caused because *WK* written in *Katakana* tended to be extracted due to the characteristic of the proposed method.

In the successful example of "*Hadekon,*" a character "o" came after the *WK*. Because the character of "o" is rarely used in *WK*, "o" becomes a clue to recognize the segment between *WK* and other characters or words. However, as in the failed example, when *Katakana* characters came after the *WK* such as "*HadekonBanzai*," the *Katakana* character string after the *WK* tended to be recognized as *WK*.

In the character strings that were extracted wrongly as *WK*, there were some strings that were easy to guess the original *WK* such as "*Inpar*" and also some strings that included *WK* internal but also included unnecessary words or characters such as "*Puchipuracheak*." These errors were also caused by the script type used after the *WK*.

In future study, it is necessary to carefully analyze these partially matched character strings and find out a clue to improve the extraction. When the sentences including *WK* are analyzed by Mecab, analysis errors such as "*Niko/Nama*" occur in Table 6. The proposed method successfully extracted the *WK* consisting of several script types such as "*Nikonama*" and proved its effectiveness.

Table 7 indicates the frequency of each combination pattern of script types when the training data size was 500. The values in brackets indicate the recall rate R_e that was calculated for each combination pattern of script types (Eq.4).

$$R_e = \frac{\text{extracted number of } WK}{\text{number of } WK \text{ included in test data}} \tag{4}$$

Table 7. The frequency of each combination pattern of script types in each method when the number of the training data was 500.

Pattern	C(%)	C+K(%)	C+K+S(%)
Kata.	17(0.172)	40(0.404)	42(0.424)
Hira.	2(0.042)	5(0.104)	5(0.104)
Kata. + Hira.	21(0.538)	29(0.744)	29(0.744)
Kata. + Chi.	6(0.182)	5(0.152)	9(0.273)
Chi.	0(0.000)	0(0.000)	2(0.087)
Kata. + Sym.	4(0.235)	5(0.294)	8(0.471)

Comparing the results by "C" and "C+K", the number of the extracted *WK* written in *Katakana* increased from 17 to 40. There was also improvement in the result when *WK* was written in both *Katakana* and *Hiragana*.

However, no clear difference was shown between "C+K" and "C+K+S". In "C" and "C+K" where stroke counts were not considered, *WK* written in alphabet character were successfully extracted, however, in "C+K+S," such *WK* were failed to be extracted. However, in "C+K+S," the results were improved in extracting *WK* including Chinese character, compared to the results in "C" and in "C+K."

6. Conclusions

In this paper, we proposed a method to extract *Wakamono Kotoba* from conversational sentences based on the features of script types and stroke counts by using CRF. As the result, the precision of the proposed method was over 70%, and proved that the method can extract *Wakamono Kotoba* in high accuracy.

However, our purpose is to extract unknown *Wakamono Kotoba* as many as possible. In future, to improve recall rate, it is necessary to consider the sense of character and the pronunciation of character as feature.

Acknowledgements

This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 23700252.

References

1. Yonekawa, A.: Wakamonogo wo kagakusuru. Meijishoin, (1998)

2. Matsuo, T. and Ando K.: Extraction of Wakamono Kotoba from Web Using Template . In: FIT2012, (2012)
3. Mori, S. and Nagao, M.: Unknown Word Extraction from Corpora Using n-gram Statistics. Transactions of Information Processing Society of Japan, Vol.39, No.7, pp.2093–2100, (1998)
4. Asahara, M. and Matsumoto, Y.: Japanese Unknown Word Identification by Character-based Chunking. In Proceedings of COLING2004, pp.459–465, (2004)
5. Ling, G. C., Asahara, M. and Matsumoto, Y.: Chinese Unknown Word Identification Using Character-based Tagging and Chunking. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL'03)- Vol.2, pp.197–200, (2003)
6. Ritter, A., Clark, S. Mausam, Etzioni, O.: Named Entity Recognition in Tweets: an Experimental Study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11), pp.1524–1534, (2011)
7. Tsuchiya, S., Imono, M., Yoshimura, E. and Watabe, H.: Meaning Judgment Method for Alphabet Abbreviation Using the Association Mechanism. In Proceedings of 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2012), pp.209–218, (2012)
8. Amiri, H. and Chua, T.-S.: Mining Slang and Urban Opinion Words and Phrases from cQA Services: an Optimization Approach. In Proceedings of the fifth ACM international conference on Web search and data mining (WSDM'12), pp.193–202, (2012)
9. Hassan, T., Soliman, A. Ali, M. A.: Mining Social Networks' Arabic Slang Comments. In Proceedings of IADIS European Conference on Data Mining 2013 (ECDM'13), pp.22–24, (2013)
10. Murawaki, Y. and Kurohashi, S.: Semantic Classification of Automatically Acquired Nouns Using Lexico-syntactic Clues. In Proceedings of 23rd International Conference on Computational Linguistics (COLING2010), pp.876–884, (2010)
11. Matsumoto, K. and Ren, F.: Construction of Wakamono Kotoba Emotion Dictionary and Its Application. In: Proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing2011), pp.405–416, (2011)
12. Matsumoto, K., Konishi, Y., Sayama, H. and Ren, F.: Analysis of Wakamono Kotoba Emotion Corpus and Its Application in Emotion Estimation, International Journal of Advanced Intelligence, Vol.3, No.1, pp.1–24, (2011)
13. Matsumoto, K., Kita, K. and Ren, F.: Emotion Estimation from Sentence Using Relation between Japanese Slangs and Emotion Expressions, In: Proceedings of 26th Pacific Asia Conference on Language,Information and Computation(PACLIC2012), pp.377–384, (2012)
14. Matsumoto, K., Kita, K. and Ren, F.: Emotional Vector Distance Based Sentiment Analysis of Wakamono Kotoba, China Communications, Vol.9, No.3, pp.87–98, (2012)
15. Nakazawa, T.: `http://nlp.ist.i.kyoto-u.ac.jp/member/nakazawa/pubdb/other/MIRA.pdf`, (2003)
16. McDonald, R., Pereira, F., Kulick, S. Winters, S. Jin, Y. and Pete, W.: Simple algorithms for complex relation extraction with applications to biomedical IE, In: Proceedings of the 43rd Annual Meeting of the Association for Compurarional Linguistics(ACL'05), pp.491–498, (2005)